

A Nearly-Quadratic Gap Between Adaptive and Non-Adaptive Property Testers

Jeremy Hurwitz*

Department of Computing and Mathematical Sciences, California Institute of Technology, Pasadena California 91125
jhurwitz@cs.caltech.edu

Abstract. We show that for all integers $t \geq 8$ and arbitrarily small $\epsilon > 0$, there exists a graph property Π (which depends on ϵ) such that ϵ -testing Π has non-adaptive query complexity $Q = \tilde{\Theta}(q^{2-2/t})$, where $q = \tilde{\Theta}(\epsilon^{-1})$ is the adaptive query complexity. This resolves the question of how beneficial adaptivity is, in the context of proximity-dependent properties ([9]). This also gives evidence that the canonical transformation of Goldreich and Trevisan ([8]) is essentially optimal when converting an adaptive property tester to a non-adaptive property tester. To do so, we provide optimal adaptive and non-adaptive testers for the combined property of having maximum degree $O(\epsilon N)$ and being a *blow-up collection* of an arbitrary base graph H .

Keywords: Sublinear-Time Algorithms, Property Testing, Dense-Graph Model, Adaptive vs Non-adaptive Queries, Hierarchy Theorem

1 Introduction

In this paper, we consider the power of adaptive versus non-adaptive queries for property testers in the dense-graph model. In this model, the algorithm is given access to the input graph $G = ([N], E)$ via an oracle $g : [N] \times [N] \rightarrow \{0, 1\}$ such that $g(u, v) = 1$ if and only if $(u, v) \in E$. A graph is said to be ϵ -far from a particular property if at least ϵN^2 edges must be added or deleted to yield a graph with the desired property.

Given a graph property Π , we say that a (randomized) algorithm \mathcal{A} is an ϵ -tester for Π if $\Pr[\mathcal{A}(G) = \text{ACCEPT}] > 2/3$ for all $G \in \Pi$ and $\Pr[\mathcal{A}(G) = \text{REJECT}] > 2/3$ for all G which are ϵ -far from Π .

1.1 Adaptive, Non-Adaptive, and Canonical Testers

Property testers can be broadly classified into two types according to how the queries are determined. In an *adaptive algorithm*, the results of previous queries can be used when determining the next query. A *non-adaptive algorithm*, on the other hand, determines all of its queries in advance.

Most of the initial work in the dense-graph model focused on non-adaptive algorithms. In fact, these early algorithms used an even more restricted framework, termed *canonical algorithms* by [8]. In a canonical algorithm, the tester chooses a random subset of the vertices and queries the entire induced subgraph.

* Supported by NSF grants CCF-0830787 and CCF-0829909

Given a property Π and an error parameter $\epsilon > 0$, let q be the adaptive query-complexity, Q be the non-adaptive query complexity, and \tilde{Q} be the canonical query complexity. By definition, $q \leq Q \leq \tilde{Q}$. A natural question, then, is to determine the exact relationship between these parameters.

In [8] and [2], the authors note that any algorithm which makes q queries considers at most $2q$ vertices. It therefore suffices to query the subgraph induced by a random set of $2q$ vertices and then locally simulate the adaptive algorithm. This *canonical transformation* shows that, for any property, $\tilde{Q} < 2q^2$.

It is easy to see that there exist properties with $\tilde{Q} = \Omega(Q^2)$,¹ thereby showing that the canonical transformation is optimal when converting (adaptive or non-adaptive) algorithms into canonical algorithms. Indeed, [5] and [7] show that a quadratic performance gap exists between non-adaptive and canonical testers for many natural properties.

Similarly, we can consider the relative power of adaptive versus non-adaptive (but not necessarily canonical) property testers. It is widely believed that the canonical transformation remains optimal, in the worst case, even for this more modest goal. In other words, it is believed that there exist properties such that $Q = \Omega(q^2)$. However, proving such a separation between the adaptive and non-adaptive complexities has proven elusive – no unconditional gap was known until Goldreich and Ron demonstrated a property where $Q = \tilde{\Omega}(q^{3/2})$ in [7].

In light of this difficulty, researchers have considered two modifications of the problem. The first approach, used by [9], considers *proximity-dependent* properties which depend on the tolerance parameter ϵ . In particular, they considered the combined property of being bipartite and having degree at most $O(\epsilon N)$, and achieved a gap of $Q = \tilde{\Theta}(q^{-4/3})$.

The second approach, used by [7], is to consider *promise problems*. In this context, the authors showed that there exist properties such that $Q = \Omega(q^{-2+\delta})$ for all $\delta > 0$, exhibiting the first nearly-quadratic gap between adaptive and non-adaptive queries in the dense-graph model. More specifically, the authors demonstrated a hierarchy of gaps of the form $Q = \Theta(q^{-2+2/t})$ for each integer $t \geq 2$. Unfortunately, the promise they use, while natural, is quite strong, and it is currently unclear how to remove the promise.

In this paper, we achieve a nearly-quadratic gap without using a promise, at the expense of making the properties proximity-dependent. This proves that for all $\delta > 0$ and arbitrarily small ϵ , there exists a property Π (which depends on ϵ), such that ϵ -testing Π requires $Q > q^{2-\delta}$ non-adaptive queries, where q is the adaptive query complexity. As in [7], we also strengthen the result and establish a hierarchy of relationships between the adaptive and non-adaptive query complexities.

Theorem 1 (Main Theorem). *For all $t \geq 8$ and arbitrarily small ϵ , there exists a graph property Π (which depends on ϵ) such that ϵ -testing Π has non-adaptive query complexity $Q = \tilde{\Theta}(q^{2-2/t})$, where $q = \tilde{\Theta}(\epsilon^{-1})$ is the adaptive query complexity.*

¹ Let Π consist of only the empty graph. Then $Q = O(\epsilon^{-1})$, while $\tilde{Q} = \Omega(\epsilon^{-2})$.

Theorem 1 and [7] both provide strong evidence that the canonical transformation is optimal in the general case. Although each result technically leaves open the possibility of a better transformation between adaptive and non-adaptive testers, each does so in a different, and very restricted, way. As a result, any such transformation would have to be very unnatural and would have to depend sensitively on the internal structure of the adaptive tester.

1.2 Graph Blow-Ups and Blow-Up Collections

A *graph blow-up* consists of replacing each vertex of a graph with a cluster of vertices (a rigorous definition is given in section 2). This operation is frequently used in studying the dense-graph model (see, for example, [1], [3], and [6]).

The complexity of testing whether a graph is a blow-up of a fixed graph H was essentially resolved in [4] (see also [5]), where it is shown that, for any H , the adaptive query complexity is $O(\epsilon^{-1})$ and the non-adaptive query complexity is $\tilde{O}(\epsilon^{-1})$.

A graph is a *blow-up collection* if it can be partitioned into disjoint subgraphs, each of which is a blow-up of H . This notion was implicitly introduced in [7], which showed the following lower bound.

Lemma 2 ([7], Lemma 5.6). *Let H be a simple t -cycle, with $t \geq 4$. Testing whether G is a blow-up collection of H requires $\Omega(\epsilon^{-(2-2/t)})$ non-adaptive queries, even given the promise that G has maximum degree $2\epsilon N$.*

In this paper, we prove tight upper-bounds for both the adaptive and non-adaptive cases. Specifically, we show that a tester can determine whether G is a blow-up collection of any given H , given that same promise on the degrees, using only $O(\epsilon^{-1})$ adaptive queries or $O(\epsilon^{-2+1/(\Delta+2)} + \epsilon^{-2+2/W})$ non-adaptive queries, where Δ and W are parameters depending only on H . When H is a simple t -cycle, $\Delta = 2$ and $W = t$, and the non-adaptive upper bound reduces to $O(\epsilon^{-2+2/t})$, matching the lower bound in Lemma 2.

Indeed, it suffices for G to be $O(\epsilon)$ -close to satisfying the promise. Since [9] gives an efficient non-adaptive tester for the property of having maximum degree $O(\epsilon N)$ ([9]), we obtain the following two theorems.

Theorem 3 (Adaptive Tester). *For all graphs H and constants $c > 1$, there exists an adaptive property tester (with two-sided error) for the (proximity-dependent) combined property of having maximum degree $c\epsilon N$ and being a blow-up collection of H . The tester has query complexity $O(\epsilon^{-1} \lg^3 \epsilon^{-1})$.*

Since any tester must make $\Omega(\epsilon^{-1})$ queries, Theorem 3 is optimal up to a polylogarithmic factor.

Theorem 3, combined with Lemma 2, suffices to show a gap of size $Q = \Omega(q^{-2+\delta})$ for all $\delta > 0$. In the following theorem, we strengthen this result by proving a tight upper-bound on the non-adaptive query complexity. This shows that there is an infinite hierarchy of achievable relationships between the adaptive and non-adaptive query complexities of proximity-dependent properties.

Theorem 4 (Non-Adaptive Tester). *For all graphs H and constants $c > 1$, there exists a non-adaptive property tester (with two-sided error) for the (proximity-dependent) combined property of having maximum degree $c\epsilon N$ and being a blow-up collection of H . The tester has query complexity $O(\epsilon^{-2+1/(\Delta+2)} + \epsilon^{-2+2/W})$, where $\Delta = \deg(H)$ is the maximum degree of H and $W < |H|^2$ is a bound on the size of a witness against H (see Definition 9).*

As mentioned previously, when H is a simple t -cycle, $\Delta = 2$ and $W = t$. Therefore, combining Theorems 3 and 4 with Lemma 2 yields Theorem 1.

We note that the algorithms in both theorems have running time polynomial in the query complexity. In the adaptive case, the query complexity can also be made polynomial in c and $|H|$. However, we do not do so here, opting instead to simplify the proof.

1.3 Organization of The Paper

Section 2 contains basic definitions and notation. In section 3, we show how to adaptively test whether a graph is a blow-up collection, given the promise that the graph is close to having maximum degree $c\epsilon N$. In section 4, we present the non-adaptive algorithm, under the same promise. Finally, section 5 removes the promise by explicitly testing that the input is close to having low degree.

2 Notation and Basics

All graphs are assumed to be undirected. Following standard graph-theoretic notation, we let $\Gamma(v) = \{u : (u, v) \in E\}$ denote the neighbors of v . Given $S \subset V$, we let $G|_S$ denote the subgraph induced by S and $\Gamma_S(v) = \Gamma(v) \cap S$ denote the neighbors of v in S . Given $S, T \subset V$, we let $E(S, T) = \{(u, v) \in E : u \in S, v \in T\}$ denote the set of edges between S and T and $S \Delta T$ denote the symmetric difference of S and T .

Definition 5 (Graph Blow-Up). *A graph $G = ([N], E)$ is a blow-up of the graph $H = ([h], F)$ if there exists a partition of $[N]$ into $V_1 \cup \dots \cup V_h$ such that for every $i, j \in [h]$ and $(u, v) \in V_i \times V_j$, $(u, v) \in E$ if and only if $(i, j) \in F$. We denote the set of blow-ups of H by $\mathcal{BU}(H)$.*

Note that no requirement is made as to the relative sizes of the V_j . In particular, we allow the case where $|V_j| = 0$.

Definition 6 (Blow-Up Collection). *A graph $G = ([N], E)$ is a blow-up collection of the graph H if there exists a partition of $[N]$ into $V^1 \cup \dots \cup V^k$, for some k , such that $G|_{V^i} \in \mathcal{BU}(H)$ for all i and $E(V^i, V^j) = \emptyset$ for all $i \neq j$. We denote the set of blow-up collections of H by $\mathcal{BUC}(H)$.*

Throughout this paper, H will be an arbitrary fixed graph and $c > 1$ will be an arbitrary fixed constant. We let $h = |H|$, $\Delta = \deg(H)$, and $\mathcal{LD}_{c\epsilon} = \{G : \deg(G) \leq c\epsilon N\}$.

To prove Theorems 3 and 4, we will repeatedly use the concept of a set being (k, α) -partitionable. Informally, a vertex is (k, α) -partitionable if almost all of its

neighbors can be partitioned into k groups, such that all of the vertices within a part have essentially the same neighbors in G . The formal definition is given in Definition 8, after we introduce notation for such partitions.

Definition 7. *Given $v \in V$ and $\alpha > 0$, let $C_{v,\alpha}(u) = \{w \in \Gamma(v) : |\Gamma(w) \Delta \Gamma(u)| < \alpha \epsilon N\}$.*

In other words, $C_{v,\alpha}(u)$ consists of all vertices in $\Gamma(v)$ which have essentially the same neighbors as u . When v and α are clear from context, we will omit the subscripts and write $C(u)$ for $C_{v,\alpha}(u)$.

Definition 8 ((k, α)-Partitionable). *A vertex v is (k, α) -partitionable if there exist representatives $u_1, \dots, u_k \in \Gamma(v)$ such that $\left| \bigcup_{i=1}^k C_{v,\alpha}(u_i) \right| \geq |\Gamma(v)| - \alpha \epsilon N$.*

Note that if $G \in \mathcal{BUC}(H)$, then every vertex is $(\deg(H), 0)$ -partitionable.

Finally, we need the idea of a minimal witness against $\mathcal{BUC}(H)$. Informally, a minimal witness is a set of vertices such that the induced subgraph proves that G is not a valid blow-up collection, while any subset does not suffice.

Definition 9 (Minimal Witness). *A set $S \subseteq V$ is a minimal witness against $\mathcal{BUC}(H)$ if $G|_S \notin \mathcal{BUC}(H)$, but for any $S' \subsetneq S$, $G|_{S'} \in \mathcal{BUC}(H)$.*

Note that $G|_S \notin \mathcal{BUC}(H)$ implies that $G \notin \mathcal{BUC}(H)$. Furthermore, any minimal witness must be connected, since we could otherwise replace the witness with one of its connected components.

Given H , we let $W = W(H)$ denote the maximum size of a minimal witness. It is easy to see that $W < \frac{1}{2}h^2$. When H is a simple t -cycle, $t \geq 4$, $W = |H|$.

3 Adaptively Testing $\mathcal{BUC}(H)$ Given a Promise

We begin by showing how to adaptively test whether a graph G is in $\mathcal{BUC}(H)$, given the promise that G is $O(\epsilon)$ -close to $\mathcal{LD}_{c\epsilon}$.

The algorithm consists of two stages. The first stage (steps 2-8) tests whether most vertices are $(\Delta, 0)$ -partitionable, where $\Delta = \deg(H)$. To do so, it repeatedly selects a vertex and then attempts to find $\Delta + 1$ neighbors of that vertex which have mutually distinct neighborhoods (see Figure 1(a)). If it finds such a set of vertices, they act as a witness against being in $\mathcal{BUC}(H)$ and the algorithm can reject with certainty in step 7.

If the algorithm fails to find such a witness, then most vertices must be (Δ, α) -partitionable, where α is a small constant. This implies that the graph must be close to being a blow-up of some base graph. In other words, almost all of the vertices can be clustered such that each pair of clusters is either almost disjoint or almost forms a complete bipartite graph. The second stage of the algorithm (steps 9-16) checks if this high-level structure is consistent with $\mathcal{BUC}(H)$ (see Figure 1(b)). It does so by performing a random search in G for $W = W(H)$ steps, where each step selects a random neighbor of the previously selected vertices. If the resulting W vertices form a witness against $\mathcal{BUC}(H)$, the algorithm rejects in step 15.

Algorithm 10. ADAPTIVEBLOWUPCOLLECTIONTEST $_{H,c}(G; \epsilon)$

1: Let $\Delta = \deg(H)$, $W = W(H)$, and $\alpha = (16\Delta|H|^2)^{-1}$.

$\backslash\backslash$ Test whether most vertices are $(\Delta, 0)$ -partitionable.

2: **for** $O(c)$ iterations **do**

3: Select a random vertex v .

4: Select a random set S of $O(\Delta\alpha^{-1}\epsilon^{-1})$ vertices, and query $v \times S$.

5: Let \bar{S} be a random subset of $\Gamma_S(v)$ of size (at most) $c\Delta\alpha^{-1}$.

6: Select a random set T of $O(\Delta^2\alpha^{-1}\epsilon^{-1})$ vertices, and query $\bar{S} \times T$.

7: If the current view of G is inconsistent with $\mathcal{BUC}(H)$, REJECT. Specifically, reject if there exist $u_1, \dots, u_{\Delta+1} \in \bar{S}$ such that $\Gamma_T(u_i) \neq \Gamma_T(u_j)$ for all i, j .

8: **end for**

$\backslash\backslash$ If G has not yet been rejected, most vertices must be (Δ, α) -partitionable.

$\backslash\backslash$ Test that G has a high-level structure consistent with H .

9: **for** $O(Wc)^{O(W)}$ iterations **do**

10: Select v_1 at random, and let $U = \{v_1\}$.

11: **for** $j = 2$ **to** W **do**

12: Select a random set T_j of $O(\Delta\epsilon^{-1})$ vertices, and query $U \times T_j$.

13: If $\Gamma_{T_j}(U) = \emptyset$, **break**. Otherwise, randomly select $v_j \in \Gamma_{T_j}(U)$ and let $U = U \cup \{v_j\}$.

14: **end for**

15: If $G|_U \notin \mathcal{BUC}(H)$, REJECT.

16: **end for**

17: If the algorithm hasn't yet rejected, ACCEPT.

Algorithm 10 has query complexity $O(\epsilon^{-1})$, as desired. Furthermore, it only rejects if it finds a witness against $\mathcal{BUC}(H)$, so the algorithm accepts valid blow-up collections with probability 1. The following lemma asserts that it rejects graphs which are far from $\mathcal{BUC}(H)$ with high probability.

Lemma 11. Let G be $\frac{\epsilon}{8}$ -close to \mathcal{LD}_{ce} and ϵ -far from $\mathcal{BUC}(H)$. Then Algorithm 10 rejects with probability at least $2/3$.

The proof is the content of section 3.1.

3.1 Proof of Lemma 11

We first note that if G contains many vertices which are not (Δ, α) -partitionable, then G is rejected with high probability.

Lemma 12. Let $\Delta = \deg(H)$, $W = W(H)$, and $\alpha = (16\Delta|H|^2)^{-1}$. Suppose that G is $\frac{\epsilon}{8}$ -close to \mathcal{LD}_{ce} and contains at least $\frac{1}{4c}N$ vertices which are not (Δ, α) -partitionable. Then Algorithm 10 rejects in step 7 with probability at least $2/3$.

The proof follows immediately from the definition of (Δ, α) -partitionability, and is given in appendix A.1.

We now show that if G has at most $\frac{1}{4c}N$ vertices which are not (Δ, α) -partitionable, then G can be partitioned into components such that, for each pair of components, either the edges between them almost form a complete bipartite graph or the components are almost disjoint.

Lemma 13. *Let $\Delta = \deg(H)$ and $\alpha = (16\Delta|H|^2)^{-1}$. Suppose that G is $\frac{\epsilon}{8}$ -close to $\mathcal{LD}_{c\epsilon}$ and contains at most $\frac{1}{4c}N$ vertices which are not (Δ, α) -partitionable. Then G is $\frac{5\epsilon}{8}$ -close to a graph $\tilde{G} = (V, \tilde{E})$ for which the following holds: V can be partitioned into $\bigcup_{i,j} V_j^i \cup L$ such that*

1. $\tilde{\Gamma}(v) = \emptyset$ for all $v \in L$.
2. $E(V^i, V^{i'}) = \emptyset$ for all $i \neq i'$.
3. For all i, j , $\tilde{\Gamma}(u) = \tilde{\Gamma}(v)$ for all $u, v \in V_j^i$.
4. $|V_j^i| > \frac{\epsilon}{16\Delta}N$ for all i, j .

where $V^i = \bigcup_j V_j^i$ and $\tilde{\Gamma}(u)$ is the neighborhood of u in \tilde{G} . Furthermore, $|I_{V^i}(u) \Delta \tilde{I}_{V^i}(u)| < \alpha\epsilon N$ for all $u \in V^i$.

The proof is given in appendix A.2.

Note that the bound on $|I_{V^i}(u) \Delta \tilde{I}_{V^i}(u)|$ in the lemma implies that \tilde{G} is $\frac{\epsilon}{8}$ -close to having maximum degree $2c\epsilon N$, since G is $\frac{\epsilon}{8}$ -close to having maximum degree $c\epsilon N$ and the degree of each vertex increases by at most $\alpha\epsilon N$ when going from G to \tilde{G} . Therefore, conditions 3 and 4 imply that most of the components V_j^i are connected to at most $32c\Delta$ other components and have size at most $2c\epsilon N$.

Intuitively, these conditions allow us to view each cluster as a supernode and each bipartite graph as an edge. From this viewpoint, \tilde{G} becomes a bounded-degree graph, with maximum degree $32c\Delta$. Since each supernode contains $\Omega(\epsilon N)$ vertices (by condition 4 in the lemma), we can simulate neighbor queries by querying $O(\epsilon^{-1})$ random vertices and choosing a random neighbor. Furthermore, if \tilde{G} is $\Omega(\epsilon)$ -far from $\mathcal{BUC}(H)$ when viewed as a dense graph, the corresponding bounded-degree graph is $\Omega(1)$ -far from $\mathcal{BUC}(H)$.

To formalize this intuition, we first show that the set of witnesses against \tilde{G} covers a constant fraction of V .

Lemma 14. *Let \tilde{G} be as in Lemma 13, and suppose that \tilde{G} is $\frac{3\epsilon}{8}$ -far from $\mathcal{BUC}(H)$. Then there exist at least $(16Wc^2\epsilon)^{-1}$ distinct sets $\tilde{W}_k = \bigcup_{\ell} V_{j_k, \ell}^{i_k}$ such that $G|_{\tilde{W}_k}$ is a witness against $\mathcal{BUC}(H)$.*

The proof is given in appendix A.3.

We are now ready to show that the second half of Algorithm 10 finds a witness against G with high probability.

By Lemma 14, step 10 of the algorithm selects a vertex $v_1 \in V_{j_1}^{i_1}$ corresponding to a witness with high probability. Having chosen v_1 , we wish to bound the probability that v_2, \dots, v_W are chosen so as to form a complete witness. Recalling that any minimal witness is connected, there must exist a $v_2 \in V_{j_2}^{i_2}$ which extends the witness in \tilde{G} . By condition 3 of Lemma 13, any vertex in $V_{j_2}^{i_2}$ can

be used in place of v_2 to extend the witness. However, since $|V_{j_2}^i| > \frac{\epsilon}{16\Delta}N$ and $|\Gamma_{V^i}(u) \Delta \tilde{\Gamma}_{V^i}(u)| < \alpha\epsilon N < \frac{1}{2W} |V_{j_2}^i|$, at least half of $V_{j_2}^i$ is a valid choice to extend the witness in G . Furthermore, most of these must have degree $O(c\epsilon N)$.

Iterating this procedure, we see that there are always at least $\Omega(\frac{\epsilon}{\Delta}N)$ choices for v_j which extend the witness in G . Furthermore, since each previous $v_{j'}$ was chosen to have degree $O(c\epsilon N)$, there are at most $O(Wc\epsilon N)$ potential choices for v_j . Therefore, with probability $\Omega((W^2c)^{-1})$, Algorithm 10 chooses a good v_j at each step and finds a complete witness in G .

The formal proof is given in Appendix A.4.

4 Non-adaptively Testing $\mathcal{BUC}(H)$ Given a Promise

We now show how to non-adaptively test whether a graph is in $\mathcal{BUC}(H)$, given the promise that G is $O(\epsilon)$ -close to $\mathcal{LD}_{c\epsilon}$.

As in the adaptive case, the first stage of the algorithm verifies that most vertices are (Δ, α) -partitionable. Assuming the graph passes the first stage, the second stage checks that the high-level structure of G is consistent with $\mathcal{BUC}(H)$.

Since we can no longer adaptively restrict our queries to neighbors of a given vertex, we instead rely on the birthday paradox to achieve sub-quadratic query complexity. Recall that the birthday paradox says that given any distribution over a discrete domain D , $O(|D|^{1-1/k})$ samples suffice to obtain a k -way collision.

Lemma 15 (Birthday Paradox). *Let \mathcal{D} be a finite domain, and let $\{\mu_i\}$ be a set of probability distribution over \mathcal{D} with $\mu_i(d) = \Omega(1/|\mathcal{D}|)$ for all i and $d \in \mathcal{D}$. Suppose that the i -th sample is drawn according to μ_i . Then $O(|\mathcal{D}|^{1-1/k})$ samples suffice to obtain a k -way collision with high probability.*

Very informally, a collision will correspond to choosing multiple vertices from a single witness in such a way that a W -way collision corresponds to a complete witness. Assuming that the set of witnesses has size $O(\epsilon^{-1})$ and that we can sample from that set with only constant overhead, the birthday paradox implies that $O(\epsilon^{-1+1/W})$ random vertices suffice to find all W vertices corresponding to a complete witness. Querying the entire induced subgraph then yields the desired result. The primary challenge, therefore, is to show that when G is ϵ -far from $\mathcal{BUC}(H)$, the algorithm can efficiently sample from the space of witnesses.

Algorithm 16. NONADAPTIVEBLOWUPCOLLECTIONTEST $_{H,c}(G; \epsilon)$

- 1: Let $\Delta = \deg(H)$, $W = W(H)$, and $\alpha = (16\Delta|H|^2)^{-1}$.
 $\backslash\backslash$ Check that most vertices are $(\Delta, 0)$ -partitionable.
- 2: Select a set S of $O((\alpha\epsilon)^{-1+1/(\Delta+2)})$ random vertices, and query $S \times S$.
- 3: Select a set T of $O(\Delta^2\alpha^{-1}\epsilon^{-1})$ random vertices, and query $S \times T$.
- 4: If the current view of G is inconsistent with $\mathcal{BUC}(H)$, REJECT. Specifically, reject if there exist $v \in S$ and $u_1, \dots, u_{\Delta+1} \in \Gamma_S(v)$ such that $\Gamma_T(u_i) \neq \Gamma_T(u_j)$ for all i, j .

$\backslash\backslash$ Check that G has a high-level structure consistent with H .

- 5: Select a set S of $O((\Delta c^2 \epsilon)^{-1+1/W})$ random vertices and query $S \times S$.
- 6: If $G|_S \notin \mathcal{BUC}(H)$, REJECT.
- 7: If the algorithm hasn't yet rejected, ACCEPT.

Algorithm 16 has query complexity $O(\epsilon^{-2+1/(\Delta+2)} + \epsilon^{-2+2/W})$, as desired. Since it only rejects if it finds a witness against G , it accepts all graphs in $\mathcal{BUC}(H)$ with probability 1. It remains to show that it rejects graphs which are far from $\mathcal{BUC}(H)$ with high probability.

Lemma 17. *Let $\alpha = (16\Delta|H|^{-2})^{-1}$, as in Algorithm 16, and let G be $\frac{\alpha\epsilon}{16c}$ -close to \mathcal{LD}_{ce} and ϵ -far from $\mathcal{BUC}(H)$. Then Algorithm 16 rejects with probability at least $2/3$.*

The proof is given in appendix B.

5 Removing the Low-Degree Promise

We now show how to test the combined property of being a valid blow-up collection and having low-degree.

In [9], the authors give an $\tilde{O}(\epsilon^{-1})$ -query algorithm for testing whether the input has maximum degree $O(\epsilon N)$.

Lemma 18 ([9], Theorem 3). *Fix $c > 1$ and $\beta > 0$. There exists a non-adaptive tester with query complexity $\tilde{O}(\epsilon^{-1})$ and two-sided error which accepts graphs with maximum degree ceN with probability at least $2/3$ and rejects graphs which are $\beta\epsilon$ -far from having maximum degree ceN with probability at least $2/3$.*

To test whether G is in $\mathcal{BUC}(H) \cap \mathcal{LD}_{ce}$, we therefore run the tester from Lemma 18, and if it accepts, we then run either the adaptive or non-adaptive tester for $\mathcal{BUC}(H)$.

All that remains is to show that if G is ϵ -far from $\mathcal{BUC}(H) \cap \mathcal{LD}_{ce}$, then it must be $\Omega(\epsilon)$ -far from $\mathcal{BUC}(H)$ or $\Omega(\epsilon)$ -far from \mathcal{LD}_{ce} .

Lemma 19. *Suppose that G is $\frac{\epsilon}{18c\Delta^2}$ -close to \mathcal{LD} and $\frac{\epsilon}{3}$ -close to $\mathcal{BUC}(H)$. Then G is ϵ -close to $\mathcal{LD} \cap \mathcal{BUC}(H)$.*

The proof of Lemma 19 is given in appendix C.

We now prove Theorems 3 and 4.

Proof (Theorems 3 and 4). We first run the tester from Lemma 18 with $\beta = (18c\Delta(H)^2)^{-1}$. If it accepts, we then run Algorithm 10 (in the adaptive case) or Algorithm 16 (in the non-adaptive case) with tolerance $\frac{\epsilon}{3}$.

If $G \in \mathcal{LD}_{ce} \cap \mathcal{BUC}(H)$, then the low-degree tester accepts with probability at least $2/3$ and blow-up collection tester accepts with probability 1. So G is accepted with probability at least $2/3$.

Suppose that G is ϵ -far from $\mathcal{LD} \cap \mathcal{BUC}(H)$. By Lemma 19, either G is $\frac{\epsilon}{18c\Delta^2}$ -far from \mathcal{LD} or G is $\frac{\epsilon}{3}$ -far from $\mathcal{BUC}(H)$. In the first case, the low-degree tester rejects with probability at least $2/3$. In the second case, the blow-up collection tester rejects with probability at least $2/3$, by Lemma 11 (in the adaptive case) or Lemma 17 (in the non-adaptive case). \square

6 Conclusions

We have shown that there exist proximity-dependent graph properties for which a non-adaptive tester must suffer an almost-quadratic increase in its query complexity over an adaptive tester. This shows that the canonical transformation is essentially optimal, in the worst case.

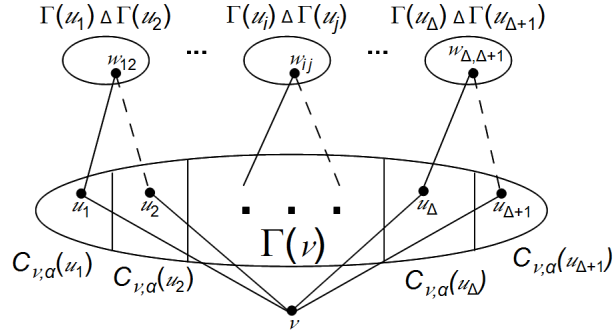
The primary open question is to remove the proximity-dependence from the graph properties used in Theorem 1. In particular, for any $\delta > 0$, does there exist a *single* graph property Π such that testing Π requires $Q = \Omega(q^{2-\delta})$ queries for all $\epsilon > 0$? It also remains to show whether there exists a nearly-quadratic separation when the adaptive algorithm is only allowed one-sided error. One approach to both of these questions is to prove an $\tilde{O}(\epsilon^{-1})$ upper-bound for the adaptive query complexity of $\mathcal{BUC}(H)$ for general graphs.

Finally, we reiterate the intriguing question raised in [7] as to what relationships are possible between the adaptive and non-adaptive query complexities. Specifically, do there exist properties such that $Q = \tilde{\Theta}(q^{2-\delta})$, with $\delta \neq \frac{2}{t}$? In particular, is it true that Q must either be $\tilde{\Theta}(q)$ or $\tilde{\Omega}(q^{4/3})$?

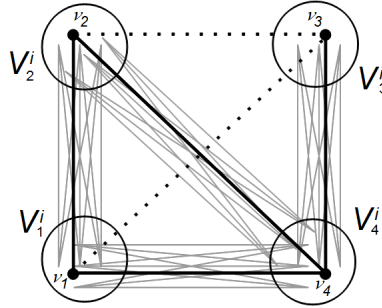
Acknowledgments: Thank you to Oded Goldreich and Chris Umans for very helpful comments and discussions about early versions of this work.

References

1. Alon, N.: Testing subgraphs in large graphs. *Random Struct. Algorithms* 21, 359–370 (October 2002)
2. Alon, N., Fischer, E., Newman, I., Shapira, A.: A combinatorial characterization of the testable graph properties: it’s all about regularity. In: *Proceedings of the thirty-eighth annual ACM Symposium on Theory of Computing*. pp. 251–260. STOC ’06, ACM, New York, NY, USA (2006)
3. Alon, N., Shapira, A.: A characterization of easily testable induced subgraphs. *Comb. Probab. Comput.* 15, 791–805 (November 2006)
4. Avigad, L.: On the lowest level of query complexity in testing graph properties. Master’s project, Weizmann Institute of Science, Department of Computer Science and Applied Mathematics (Dec 2009), <http://www.wisdom.weizmann.ac.il/~oded/msc-1a.html>
5. Avigad, L., Goldreich, O.: Testing graph blow-up (Mar 2010), http://www.wisdom.weizmann.ac.il/~oded/p_lidor.html, to appear in *RANDOM* 2011
6. Goldreich, O., Krivelevich, M., Newman, I., Rozenberg, E.: Hierarchy theorems for property testing. In: *Approximation Algorithms for Combinatorial Optimization*. pp. 504–519 (2009)
7. Goldreich, O., Ron, D.: Algorithmic aspects of property testing in the dense graphs model. *SIAM Journal on Computing* 40(2), 376–445 (2011)
8. Goldreich, O., Trevisan, L.: Three theorems regarding testing graph properties. *Random Struct. Algorithms* 23, 23–57 (August 2003)
9. Gonen, M., Ron, D.: On the benefits of adaptivity in property testing of dense graphs. *Algorithmica* 58, 811–830 (December 2010)



(a) A witness against v being $(\Delta, 0)$ -partitionable consists of two sets of vertices. The first set consists of $\Delta + 1$ vertices $u_1, \dots, u_{\Delta+1} \in \Gamma(v)$. The second set consists of $\binom{\Delta+1}{2}$ vertices w_{ij} such that $w_{ij} \in \Gamma(u_i) \Delta \Gamma(u_j)$ for each $i \neq j$. When v is not (Δ, α) -partitionable, there exist $\alpha \epsilon N$ choices for each u_i such that $u_i \notin C_{v,\alpha}(u_j)$ for any $j \neq i$, so Algorithm 10 can find such vertices efficiently. Having chosen such a set, there exist $\alpha \epsilon N$ choices for w_{ij} for each $i \neq j$, allowing algorithm 10 to complete the witness efficiently. Steps 2-8 search for witnesses of this type.



(b) If G has not been rejected by the first half of Algorithm 10, then V can be partitioned into clusters V_j^i , such that each pair of clusters either nearly forms a complete bipartite graph or is nearly disjoint. A minimal witness against having the correct high-level structure consists of a representative vertex from each set such that the induced subgraph is inconsistent with $\mathcal{BUC}(H)$. For example, when $H = K_4$ is the complete graph on 4 vertices, the four vertices shown here form a minimal witness. Steps 9-16 search for witnesses of this type.

Fig. 1: The two types of witnesses used by Algorithm 10 to determine whether G is a member of $\mathcal{BUC}(H)$. In each diagram, a line indicates that the algorithm queried that edge, with a solid line indicating the presence of an edge and a dashed line indicating the absence of an edge.

Appendix

A Formal Proofs for Section 3

A.1 Proof of Lemma 12

Consider the $\frac{1}{4c}N$ vertices which are not (Δ, α) -partitionable. By assumption, at most $\frac{1}{8c}N$ of them have degree greater than $3c\epsilon N$. For otherwise, G would have distance at least $\frac{1}{2} \cdot 2c\epsilon N \cdot \frac{1}{8c}N = \frac{\epsilon}{8}N^2$ from \mathcal{LD} .

Therefore, with high probability, some iteration chooses a vertex v in step 3 that is not (Δ, α) -partitionable and has degree at most $3c\epsilon N$. Consider that iteration. Note that since v is not (Δ, α) -partitionable, $|\Gamma(v)| > \alpha\epsilon N$.

We view S as $S_1 \cup \dots \cup S_{\Delta+1}$, with $|S_i| = \alpha^{-1}\epsilon^{-1}$. With high probability, $\Gamma_{S_1}(v) \neq \emptyset$, so let u_1 be an arbitrary vertex in $\Gamma_{S_1}(v)$. Since v is not (Δ, α) -partitionable, $\Gamma(v)$ contains at least $\alpha\epsilon N$ vertices u' such that $|\Gamma(u_1) \Delta \Gamma(u')| > \alpha\epsilon N$, and so with high probability $\Gamma_{S_2}(v)$ contains such a vertex. Let u_2 be that vertex. We continue in this manner, selecting vertices $u_3, \dots, u_{\Delta+1}$ such that $|\Gamma(u_i) \Delta \Gamma(u_j)| > \alpha\epsilon N$ for all $i \neq j$.

If $|\Gamma_S(v)| < c\Delta\alpha^{-1}$, then $\bar{S} = \Gamma_S(v)$. Otherwise, the probability that each vertex in \bar{S} is a valid choice for u_i , given u_1, \dots, u_{i-1} , is at least $\frac{\alpha}{3c}$, so with high probability \bar{S} contains the desired $\Delta + 1$ vertices.

Viewing T as $\bigcup_{i,j=1}^{\Delta+1} T_{ij}$, we see that, with high probability, T contains a vertex in $\Gamma(u_i) \Delta \Gamma(u_j)$ for all $i \neq j$. The resulting view of G is therefore inconsistent with $\mathcal{BUC}(H)$ and the algorithm rejects, as desired. \square

A.2 Proof of Lemma 13

First, delete all edges adjacent to any vertex which is not (Δ, α) -partitionable and add those vertices to L . The total cost of doing so is at most $\frac{1}{4c}N \cdot c\epsilon N + \frac{\epsilon}{8}N^2 = \frac{3c}{8}N^2$, since G is $\frac{\epsilon}{8}$ close to $\mathcal{LD}_{c\epsilon}$.

The construction now proceeds in stages, with each stage constructing a $V^i = \bigcup_j V_j^i$ such that $\tilde{E}(V^i, V \setminus V^i) = \emptyset$.

To construct $V^1 = \bigcup_j V_j^1$, choose an arbitrary vertex v with $|\Gamma(v)| > \frac{\epsilon}{8}N$ and (Δ, α) -partition $\Gamma(v)$ into $\bigcup_{j=1}^{\Delta} C(u_j)$. For any j such that $|C(u_j)| > \frac{\epsilon}{16\Delta}N$, add $C(u_j)$ to V^1 as a distinct V_j^1 . Note that since $|\Gamma(v)| > \frac{\epsilon}{8}N$, there must be at least one partition of that size.

While there exists a u_j in V^1 such that $|\Gamma(u_j) \setminus V^1| > \frac{\epsilon}{8}N$, choose such a vertex. For consistency of notation, we relabel u_j to v . Let $\bigcup_{j=1}^{\Delta} C(u_j)$ be a (Δ, α) -partitioning of $\Gamma(v)$. Again, there must be a component such that $|C(u_j) \setminus V^1| > \frac{\epsilon}{16\Delta}N$. We add those $C(u_j) \setminus V^1$ to V^1 as distinct V_j^1 s.

Once every u_j in V^1 has $|\Gamma(u_j) \setminus V^1| < \frac{\epsilon}{8}N$, we are ready to finalize V^1 . We begin by separating V^1 from $V \setminus V^1$ by deleting $E(V^1, V \setminus V^1)$, at a cost per vertex of at most $\frac{\epsilon}{8}N + \alpha\epsilon N$. We next ensure that $E(V_j^1, V_{j'}^1)$ is either $V_j^1 \times V_{j'}^1$ or the empty set, for all j, j' . This can be easily shown to require at most $\alpha\epsilon N$ edits per vertex.

Note that V^1 now satisfies conditions 2-5, by construction.

We now repeat this entire process on the remaining vertices, creating V^2, V^3, \dots , until every vertex in V has either been covered or has degree at most

$\frac{\epsilon}{8}N$. Finally, we delete all edges adjacent to the leftover vertices, at a cost of $\frac{\epsilon}{8}N$ per vertex, and add those vertices to L .

The total cost of this procedure is bounded by $(\frac{\epsilon}{8}N + 2\alpha\epsilon N)N < \frac{\epsilon}{4}N^2$, for a final cost of $\frac{5\epsilon}{8}N^2$, as desired. \square

A.3 Proof of Lemma 14

First, delete $\frac{\epsilon}{8}N^2$ edges so that \tilde{G} is in $\mathcal{LD}_{2c\epsilon}$. Note that the resulting graph is still $\frac{\epsilon}{4}$ -far from $\mathcal{BUC}(H)$.

Consider some \tilde{W}_k . Completely disconnecting all vertices contained in \tilde{W}_k requires at most $W(2c\epsilon N)^2$ deletions, since there are at most W components and, by Lemma 13, each component contains at most $2c\epsilon N$ vertices and each vertex has degree at most $2c\epsilon N$.

Therefore, if there are less than $(16Wc^2\epsilon)^{-1}$ disjoint \tilde{W}_k , the total cost of deleting all witnesses in \tilde{G} is at most $(16Wc^2\epsilon)^{-1} \cdot W(2c\epsilon N)^2 = \frac{\epsilon}{4}N^2$. This contradicts the assumed distance to $\mathcal{BUC}(H)$, and the lemma follows. \square

A.4 Proof of Lemma 11

If G is rejected with probability at least $2/3$ by the first part of Algorithm 10, then we are done. So suppose otherwise. Then by Lemma 12, there exists \tilde{G} satisfying the conclusion of Lemma 13. Since G is ϵ -far from $\mathcal{BUC}(H)$, \tilde{G} is at least $\frac{3\epsilon}{8}$ -far from $\mathcal{BUC}(H)$.

By Lemma 14, there exist $(16Wc^2\epsilon)^{-1}$ sets $\tilde{W}_k = \bigcup_{\ell} V_{j_k, \ell}^{i_k}$ corresponding to witnesses. Call a \tilde{W}_k *high-degree* if, for some $V_{j_k, \ell}^{i_k} \subset \tilde{W}_k$ contains $\frac{\epsilon}{64\Delta}N$ vertices of degree greater than $65\Delta Wc^2\epsilon N$. We note that at most half of the \tilde{W}_k are high-degree. Otherwise, G would have distance at least $\frac{1}{2} \cdot (16Wc^2\epsilon)^{-1} \cdot 64\Delta Wc^2\epsilon N \cdot \frac{\epsilon}{64\Delta}N = \frac{\epsilon}{32}N^2$ to $\mathcal{LD}_{c\epsilon}$, contrary to assumption.

We therefore restrict our attention to the $(32Wc^2\epsilon)^{-1}$ witnesses which are not high-degree.

Let $V_1^i, \dots, V_\ell^i, \ell < W$, be components corresponding to a partial witness in \tilde{G} , and let v_1, \dots, v_ℓ be arbitrary vertices in V_1^i, \dots, V_ℓ^i , respectively, such that $|\Gamma(v_j)| < 65\Delta Wc^2\epsilon N$ for each $j \leq \ell$. Let $V_{\ell+1}^i$ be a component adjacent to $\bigcup_{j=1}^{\ell} V_j^i$ which extends the partial witness in \tilde{G} . Such a component must exist, since witnesses are connected.

Since $|V_{\ell+1}^i| > \frac{\epsilon}{16\Delta}N$, there are at least $\frac{\epsilon}{16\Delta}N$ vertices which progress the witness (in \tilde{G}). Recall that if V_j^i is adjacent to $V_{j'}^i$ in \tilde{G} , then every vertex in V_j^i must be adjacent in G to all but at most $\alpha\epsilon N$ vertices in $V_{j'}^i$ (and similarly if V_j^i is not adjacent to $V_{j'}^i$). Therefore, since $W < \frac{1}{2}|H|^2$ and $\alpha = (16\Delta|H|^2)^{-1}$, there must be at least $\frac{\epsilon}{16\Delta}N - W\alpha\epsilon N > \frac{\epsilon}{32\Delta}N$ choices for $v_{\ell+1} \in V_{\ell+1}^i$ such that $E(v_j, v_{\ell+1}) = \tilde{E}(v_j, v_{\ell+1})$ for all $j \leq \ell$, which means that $v_{\ell+1}$ progresses the witness in G as well. Of these, at least $\frac{\epsilon}{64\Delta}N$ must also have degree at most $65\Delta Wc^2\epsilon N$.

Since $|T_i| = O(\Delta^{-1}\epsilon^{-1})$, with high probability $\Gamma_{T_i}(U)$ contains such a vertex in step 13. Since each of the ℓ vertices selected so far is adjacent to at most $65c\epsilon N$

vertices, there are at most $\ell 65c\epsilon N$ candidates for $v_{\ell+1}$, and so the probability that the selected $v_{\ell+1}$ progresses the witness in G is at least $\frac{\epsilon/32\Delta}{\ell 65c\epsilon} = \Omega((W^2c)^{-1})$.

Therefore, with high probability, the algorithm finds a complete witness in some iteration of steps 10-14 and rejects G . \square

B Proof of Correctness for the Non-Adaptive Tester

As in the proof for the adaptive algorithm, we first show that if G contains many vertices which are not (Δ, α) -partitionable, then G is rejected with high probability.

Lemma 20. *Let $\Delta = \deg(H)$, $W = W(H)$, and $\alpha = (16\Delta|H|^2)^{-1}$. Suppose that G is $\frac{\alpha\epsilon}{16c}$ -close to $\mathcal{LD}_{c\epsilon}$ and contains at least $\frac{1}{4c}N$ vertices which are not (Δ, α) -partitionable. Then Algorithm 16 rejects in step 4 with probability at least $2/3$.*

For ease of exposition, we begin by proving the claim under the stronger assumption that $G \in \mathcal{LD}_{c\epsilon}$. We then show how to modify the argument to only require that G be $O(\epsilon)$ -close to $\mathcal{LD}_{c\epsilon}$.

Proof. We view the algorithm as choosing S one vertex at a time. As the samples are chosen, we maintain an approximate partitioning of V into $\Omega(\epsilon^{-1})$ sets $\bigcup_k U_k \cup L$, such that $|U_k| > \alpha\epsilon N$ for all k and no vertex belongs to more than Δ of the U_k . For each k , we initially say that U_k is *uninitialized*. The first time that we select $v \in U_k$, we say that U_k is *initialized with seed v* , after which we require that $U_k \subseteq \Gamma(v)$.

Initially, let L consist of all (Δ, α) -partitionable vertices. Arbitrarily divide the remaining $\frac{1}{4c}N$ vertices evenly between the U_k .

Whenever $u \in U_k$ is chosen, we update the partitioning as follows. If U_k is uninitialized, then u must not be (Δ, α) -partitionable (by construction). We associate U_k with u and set $U_k = \Gamma(u)$. Note that since u is not (Δ, α) -partitionable, $|U_k| > \alpha\epsilon N$ as required. We also update the remaining uninitialized $U_{k'}$ by removing any vertex $v \in U_{k'}$ such that $v \in \Gamma(u)$ or $|\Gamma(u) \Delta \Gamma(v)| < \frac{1}{2}\alpha\epsilon N$. Finally, we rebalance the sizes of the uninitialized $U_{k'}$. Supposing for the moment that $G \in \mathcal{LD}_{c\epsilon}$, we remove at most $|\Gamma(u)| \cdot c\epsilon N / \frac{1}{2}\alpha\epsilon N < 2c^2\alpha^{-1}\epsilon N$ vertices.

If vertices v, u_1, \dots, u_ℓ , $\ell \leq \Delta$, have been chosen from U_k , let

$$U_k = \Gamma(v) \setminus \bigcup_{i=1}^{\ell} C_{v,\alpha}(u_i).$$

Note that this ensures that $u_i \in \Gamma(v)$ and $|\Gamma(u_i) \Delta \Gamma(u_j)| > \alpha\epsilon N$ for all i, j . Furthermore, since v is not (Δ, α) -partitionable, we must have $|U_k| > \alpha\epsilon N$.

We continue this way until either (i) $\Delta + 2$ samples have been chosen from a single U_k or (ii) at least half of the vertices in some initialized U_k belong to at least Δ other initialized $U_{k'}$. We claim that this occurs within $O(\epsilon^{-1+1/(\Delta+2)})$ steps.

To see this, suppose that condition (ii) has not occurred. First note that either $(8c^3\alpha^{-1}\epsilon)^{-1}$ of the U_k have been initialized, or at least $\frac{1}{8c}N$ vertices are contained in the uninitialized U_k . Therefore,

$$\left| \bigcup_k U_k \right| > (8c^3\alpha^{-1}\epsilon)^{-1} \cdot \frac{\alpha\epsilon}{2\Delta} N = \frac{\alpha^2}{8c^3} N = \Omega(N),$$

so $O(1)$ samples from V suffice to sample from U_k . By the birthday paradox, it follows that $O((\frac{\Delta}{\alpha}\epsilon)^{-1+1/(\Delta+2)}) = O(\epsilon^{-1+1/(\Delta+2)})$ samples suffice to obtain the desired $(\Delta+2)$ -way collision, with high probability.

Suppose that we halted due to having chosen $v, u_1, \dots, u_{\Delta+1}$ from some U_k . Then with high probability, T contains a vertex from $\Gamma(u_i) \triangle \Gamma(u_j)$, for each i, j , and the algorithm rejects in step 4.

If, instead, we halted due to condition (ii), then there exist $\frac{\alpha\epsilon}{2}N$ vertices which are adjacent to $\Delta+1$ of the seed vertices. With high probability, T contains such a vertex u . Let $v_1, \dots, v_{\Delta+1}$ be the corresponding seed vertices. Recall that, by construction, $|\Gamma(v_i) \triangle \Gamma(v_j)| > \frac{1}{2}\alpha\epsilon N$ for all seed vertices v_i, v_j . Therefore, with high probability, T contains a vertex in $\Gamma(v_i) \triangle \Gamma(v_j)$ for each $i \neq j$, and the algorithm rejects in step 4.

Note that in the preceding argument, we only assumed that $G \in \mathcal{LD}_{c\epsilon}$ when bounding the number of vertices discarded after each step from the uninitialized $U_{k'}$. Note, however, that each additional discarded vertex implies an additional $\frac{1}{2}\alpha\epsilon N$ distance from $\mathcal{LD}_{c\epsilon}$. Therefore, we delete at most an additional $\frac{1}{16c}N$ vertices. It follows that $|\bigcup_k U_k| = \Omega(N)$, as required, and the rest of the proof goes through unchanged. \square

By Lemma 20, either G is rejected with high probability by step 4 or we can apply Lemma 13 to obtain \tilde{G} , as in the adaptive case. By Lemma 14, it follows that there are many distinct witnesses against \tilde{G} , which allows us to again apply the birthday paradox.

Specifically, let $\mathcal{D} = \{\tilde{W}_k\}$ be the set of witnesses guaranteed by Lemma 14. As was shown in the adaptive case, for any choice of v_1, \dots, v_ℓ corresponding to some \tilde{W}_k , there exist at least $\frac{\epsilon}{32\Delta}N$ choices for $v_{\ell+1}$ which correctly extend \tilde{W}_k in G . Therefore, $O(Wc^2\Delta)$ samples suffice to obtain a vertex which extends some witness. Applying the birthday paradox, we see that step 5 finds a complete witness with high probability, in which case the algorithm rejects in step 6.

We now formalize this argument.

Proof (Lemma 17).

If G is rejected with probability at least $2/3$ by the first part of Algorithm 16, then we are done. So suppose otherwise. Then by Lemma 20, there exists \tilde{G} satisfying the conclusion of Lemma 13. So by Lemma 14, there exists at least $(8Wc^2\epsilon)^{-1}$ distinct \tilde{W}_k .

As in the proof of Lemma 11, let V_1^i, \dots, V_ℓ^i , $\ell < W$, be components corresponding to a partial witness in \tilde{G} , and let v_1, \dots, v_ℓ be arbitrary vertices in

V_1^i, \dots, V_ℓ^i , respectively. Let $V_{\ell+1}^i$ be a component adjacent to $\bigcup_{j=1}^\ell V_j^i$ which extends the partial witness in \tilde{G} . Such a component must exist, since minimal witnesses are connected.

Since $|V_{\ell+1}^i| > \frac{\epsilon}{16\Delta}N$, there are at least $\frac{\epsilon}{16\Delta}N$ vertices which progress the witness (in \tilde{G}). Also recall that if V_j^i is adjacent to $V_{j'}^i$ in \tilde{G} , then every vertex in V_j^i must be adjacent in G to all but at most $\frac{\epsilon}{16\Delta h^2}$ vertices in $V_{j'}^i$ (and similarly if V_j^i is not adjacent to $V_{j'}^i$). Therefore, since $W < \frac{1}{2}h^2$, there must be at least $\frac{\epsilon}{16\Delta}N - W \frac{\epsilon}{16\Delta h^2}N > \frac{\epsilon}{32\Delta}N$ choices for $v_{\ell+1}$ such that $E(v_j, v_{\ell+1}) = \tilde{E}(v_j, v_{\ell+1})$ for all $j \leq \ell$, which means that $v_{\ell+1}$ progresses the witness in G as well.

We now map step 5 onto the birthday paradox. Given vertices v_1^k, \dots, v_ℓ^k in $V_{j_{k,1}}^{i_k}, \dots, V_{j_{k,\ell}}^{i_k}$, respectively, let $U_k \subseteq V_{j_{k,\ell+1}}^{i_k}$ be the set of vertices which extend \tilde{W}_k in G .

We let $\mathcal{D} = \{U_k\}$. Since $|U_k| > \frac{\epsilon}{32\Delta}N$ for all k and the U_k are disjoint, $O(Wc^2\Delta)$ samples from V suffice to obtain a sample from \mathcal{D} . Therefore, by Lemma 15, $O(\epsilon^{-1+1/W})$ samples suffice to obtain a complete witness, which means that S contains a complete witness with high probability and the algorithm rejects in step 6. \square

C Proof of Lemma 19

Let $\epsilon_1 = \frac{\epsilon}{18c\Delta^2}$ be the distance to \mathcal{LD}_{cc} and $\epsilon_2 = \frac{\epsilon}{3}$ be the distance to $\mathcal{BUC}(H)$. Let $V = \bigcup V_j^i$ be the optimal decomposition of G with respect to $\mathcal{BUC}(H)$.

First, we completely disconnect all V_j^i such that $|V_j^i| < \frac{\epsilon}{3\Delta}N$. Next, we delete all superfluous edges and add all missing edges so that $G \in \mathcal{BUC}(H)$, at total cost at most $\epsilon_2 N^2$. Note that since $\bigcup V_j^i$ is the optimal decomposition, every vertex must have been connected to at least half of its neighbors. Therefore, at worst this doubled the degree of every vertex, thereby doubling the distance to \mathcal{LD} . Note that G is now a valid blow-up collection.

The total cost of the edits so far is bounded by $\epsilon_2 N^2 + \sum |V_j^i| \Delta \frac{\epsilon}{3\Delta}N = \epsilon_2 N^2 + \frac{\epsilon}{3}N^2$, since each component is adjacent to at most Δ other components.

For each i, j , if $|V_j^i| > c\epsilon N$, delete $|V_j^i| - c\epsilon N$ vertices from V_j^i . This clearly preserves membership in $\mathcal{BUC}(H)$. Furthermore, the cost of doing the deletion is exactly equal to the decrease in distance to \mathcal{LD}_{cc} . To see this, note that every neighbor of V_j^i needed to delete at least that many edges into V_j^i , and by symmetry we can assume that they all delete corresponding edges.

Finally, while any high-degree vertices remain, we do the following. First, choose $u \in V_j^i$ such that $|\Gamma(u)| > c\epsilon N$ and $v \in \Gamma(u)$, and delete v . Note that the cost of doing so is, at most, $c\epsilon N \cdot \Delta$ and the decrease in distance is at least $|V_j^i| > \frac{\epsilon}{3\Delta}N$, for a net multiplicative overhead of $3c\Delta^2$.

The total cost of removing the high-degree vertices is therefore bounded by $3c\Delta^2 \cdot 2\epsilon_1 N^2$, which means that the final cost is bounded by $(\epsilon_2 N^2 + \frac{\epsilon}{3}N^2) + (3c\Delta^2 \cdot 2\epsilon_1 N^2) = \epsilon N^2$ as desired. \square